

Ice Sheet System Model Installation and Quick Start

Eric LAROUC¹,

¹Jet Propulsion Laboratory, Pasadena, CA



Outline

1 Preliminary Notes

- System Requirements
- Required Tools
- Downloading
- License

2 ISSM Pre-Installation

- Environment Variables
- External Packages

3 ISSM Installation

- ISSM Configuration
- ISSM Compilation

4 Example: Steady state ice shelf with square domain

- Model set up
- Model parameterization
- Compute & display model solution



Outline

1 Preliminary Notes

- System Requirements

- Required Tools

- Downloading

- License

2 ISSM Pre-Installation

- Environment Variables

- External Packages

3 ISSM Installation

- ISSM Configuration

- ISSM Compilation

4 Example: Steady state ice shelf with square domain

- Model set up

- Model parameterization

- Compute & display model solution



System Requirements

- Operating System Requirements:
 - *nix systems:
 - MacOSX 32/64
 - Linux 32/64 (Debian, Ubuntu, etc .)
 - Unix/SGI
 - Windows OS: XP/Vista/7 (untested on Windows 8) – > via Cygwin.
- Software Requirements:
 - Matlab (version 8 and more recommended)
 - Python (2.7, compatible with 3.0 but not tested)
 - Java port in the making.

Required Tools

The following software will be used throughout:

- Subversion (SVN)
 - A centralized Version Control System (VCS) utilized by the ISSM team.
- The GNU Build System (autotools)
 - Responsible for ISSM's portability.
- GNU Make
 - Target driven tool used to compile ISSM.
- libtools
 - Enhanced portability

Obtaining ISSM Distribution

Download Instructions: <http://issm.jpl.nasa.gov/download/>

- Download and install SVN (Apache Subversion)
- Checkout ISSM:

```
$ svn -username anon -password anon checkout  
http://issm.ess.uci.edu/svn/issm/issm/trunk
```
- Update ISSM:

```
$ svn update
```



License

- Three-Clause BSD License:

Copyright ©2002-2012, California Institute of Technology.

All rights reserved. Based on Government Sponsored Research under contracts NAS7-1407 and/or NAS7-03001.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- 1) Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- 2) Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- 3) Neither the name of the California Institute of Technology (Caltech), its operating division the Jet Propulsion Laboratory (JPL), the National Aeronautics and Space Administration (NASA), nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE CALIFORNIA INSTITUTE OF TECHNOLOGY BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.



Outline

1 Preliminary Notes

- System Requirements
- Required Tools
- Downloading
- License

2 ISSM Pre-Installation

- Environment Variables
- External Packages

3 ISSM Installation

- ISSM Configuration
- ISSM Compilation

4 Example: Steady state ice shelf with square domain

- Model set up
- Model parameterization
- Compute & display model solution



Environment Variables

Website:

<http://issm.jpl.nasa.gov/download/unix/>

ISSM needs the following Environment Variables to be set:

- `.bashrc`
`export ISSM_DIR=ISSMPATH`
`source $ISSM_DIR/etc/environment.sh`
- `.cshrc`
`setenv ISSM_DIR ISSMPATH`
`source $ISSM_DIR/etc/environment.csh`

ISSMPATH is the path of ISSM main directory

- **ex:** `/scratch/issm/`

External Packages: Fortran Compiler

A Fortran compiler is required for some of the PETSc packages

- Mac OS X:
Xcode does Not have a Fortran compiler

- Download and install binaries:

<http://gcc.gnu.org/wiki/GFortranBinaries>

External Packages

Website:

<http://issm.jpl.nasa.gov/download/unix/>

External Packages to Install: Order Matters!

- 1 Autotools
 - Autoconf
 - Automake
 - Libtool
- 2 MPICH
 - Implements Message Passing Interface (MPI) versions one and two.
- 3 CMAKE
 - Cross-platform open source build system. Needed to compile some of the external packages
- 4 PETSc
 - A suite of data structures and routines for the scalable (parallel) solution of scientific applications modeled by partial differential equations.

External Packages

5 Triangle

- Triangle generates exact Delaunay triangulations, constrained Delaunay triangulations, conforming Delaunay triangulations, Voronoi diagrams, and high-quality triangular meshes.

6 Matlab

- Only a softlink to the actual matlab directory.

External Packages are located in the trunk folder
(i.e. `$ cd $ISSM_DIR/externalpackages/`)

A complete install of python related external packages is also possible,
instructions available at: <http://issm.jpl.nasa.gov/download/unix/>

External Packages: Installing

Typical installation process requires one to run the appropriate script.

- i.e. `install-linux64.sh` if you are running 64 bit Linux.

Installation script takes care of:

- Downloading the package tarball.
- Uncompressing and extracting.
- Configuring the package to your platform.
- Compiling and installing.

Example: PETSc

Tweaking the configure.sh file may be necessary for custom installations. For example, difference between a linux64 and a cluster install:

```

#!/bin/bash
set -eu

#Some cleanup
rm -rf install petsc-3.4.3 src
mkdir install src

#Download from ISSM server
ISSM_DIR/scripts/DownloadExternalPackage.py 'http://issm.jpl.nasa.gov/files/externalpackages/petsc-lite-3.4.3.tar.gz' 'petsc-3.4.3.tar.gz'

#Unrar and move petsc to install directory
tar -zxvf petsc-3.4.3.tar.gz
mv petsc-3.4.3/* src/
rm -rf petsc-3.4.3

#configure
cd src
./config/configure.py \
  --prefix="$ISSM_DIR/externalpackages/petsc/install" \
  --PETSC_ARCH="$ISSM_ARCH" \
  --PETSC_DIR="$ISSM_DIR/externalpackages/petsc/src" \
  --with-debugging=0 \
  --with-mpi-dir="$ISSM_DIR/externalpackages/mpi/ch/install" \
  --with-mpi-lib="$ISSM_DIR/externalpackages/mpi/lib/install" \
  --with-mpi-include="$ISSM_DIR/externalpackages/mpi/include/install" \
  --with-shared-libraries=1 \
  --download-metis=yes \
  --download-mumps=yes \
  --download-scalapack=yes \
  --download-blas=yes \
  --download-blas-yes \
  --download-fblas-lapack=yes \
  --with-p1c=1

#Compile petsc and install it
make
make install

#Some cleanup
rm -rf install petsc-3.4.3 src
mkdir install src

#Download from ISSM server
ISSM_DIR/scripts/DownloadExternalPackage.py 'http://issm.jpl.nasa.gov/files/externalpackages/petsc-lite-3.4.3.tar.gz' 'petsc-3.4.3.tar.gz'

#Unrar and move petsc to install directory
tar -zxvf petsc-3.4.3.tar.gz
mv petsc-3.4.3/* src/
rm -rf petsc-3.4.3

#configure
cd src
./config/configure.py \
  --prefix="$ISSM_DIR/externalpackages/petsc/install" \
  --PETSC_ARCH="$ISSM_ARCH" \
  --PETSC_DIR="$ISSM_DIR/externalpackages/petsc/src" \
  --with-debugging=0 \
  --with-mpi-dir="$ISSM_DIR/externalpackages/mpi/ch/install" \
  --with-mpi-lib="$ISSM_DIR/externalpackages/mpi/lib/install" \
  --with-mpi-include="$ISSM_DIR/externalpackages/mpi/include/install" \
  --with-shared-libraries=1 \
  --download-metis=yes \
  --download-mumps=yes \
  --download-scalapack=yes \
  --download-blas=yes \
  --download-blas-yes \
  --download-fblas-lapack=yes \
  --with-p1c=1

#Compile petsc and install it
make
make install

#Prepare script to reconfigure petsc
cat > script.queue << EOF
PBS -S /bin/bash
PBS -l nodes=1:ppn=1
PBS -q cis145
PBS -l walltime=10

cd $(echo $ISSM_DIR)/externalpackages/petsc/src/
./configure --prefix=$ISSM_ARCH/linux64-c-ant
EOF

#Print instructions
echo "=== How: cd src/"
echo "=== qsub script.queue"
echo "=== Then run reconfigure script generated by PETSc and follow instructions"

```

Outline

- 1 Preliminary Notes
 - System Requirements
 - Required Tools
 - Downloading
 - License
- 2 ISSM Pre-Installation
 - Environment Variables
 - External Packages
- 3 ISSM Installation
 - ISSM Configuration
 - ISSM Compilation
- 4 Example: Steady state ice shelf with square domain
 - Model set up
 - Model parameterization
 - Compute & display model solution



ISSM Configuration

Generate the makefiles needed to compile ISSM:

- ISSM uses Autotools to make the source-code packages portable to many Unix-like systems
- Preconfigure ISSM:

```
$ cd $ISSM_DIR  
$ ./scripts/automakererun.sh
```

Configure ISSM for your Operating System/Architecture

- create a configure.sh files corresponding to your machine type (several examples are provided in `$ISSM_DIR/configs`)
- For example:

```
$ cp ./configs/configure.sh configure.sh  
$ ./configure.sh
```



Custom ISSM Installation

Other platforms may require the user to write their own configure.sh file All of the options available for configuring ISSM can be listed by running \$

```
./configure -help
```

```
1  #!/bin/sh
2  ./configure \
3      --prefix=$ISSM_DIR \
4      --with-matlab-dir="$ISSM_DIR/externalpackages/matlab/install" \
5      --with-triangle-dir="$ISSM_DIR/externalpackages/triangle/install" \
6      --with-mpi-include="$ISSM_DIR/externalpackages/mpich/install/include" \
7      --with-mpi-libflags="-L$ISSM_DIR/externalpackages/mpich/install/lib/ ...
      -lmpich" \
8      --with-petsc-dir="$ISSM_DIR/externalpackages/petsc/install" \
9      --with-petsc-arch="$ISSM_ARCH" \
10     --with-metis-dir="$ISSM_DIR/externalpackages/petsc/install" \
11     --with-blas-lapack-dir="$ISSM_DIR/externalpackages/petsc/install" \
12     --with-scalapack-dir="$ISSM_DIR/externalpackages/petsc/install/" \
13     --with-mumps-dir="$ISSM_DIR/externalpackages/petsc/install/" \
14     --with-numthreads=2
```



ISSM Compilation

Compile ISSM:

- ISSM can now be compiled:

```
$ cd $ISSM_DIR  
$ make  
$ make install
```

ISSM installation is done!

Compiling Troubleshooting:

- <http://issm.jpl.nasa.gov/installation/compilationtroubleshooting/>



Outline

1 Preliminary Notes

- System Requirements
- Required Tools
- Downloading
- License

2 ISSM Pre-Installation

- Environment Variables
- External Packages

3 ISSM Installation

- ISSM Configuration
- ISSM Compilation

4 Example: Steady state ice shelf with square domain

- Model set up
- Model parameterization
- Compute & display model solution



Example: Steady state ice shelf with square domain

- Go to working directory

```
1 $ cd $ISSM_DIR/examples/SquareIceShelf/
```

- Start-up Matlab

```
1 $ matlab
```

- Make Matlab aware of ISSM

```
1 >> addpath ISSM_DIR/bin  
2 >> addpath ISSM_DIR/lib
```

- In Matlab, create empty model structure

```
1 >> md=model;
```

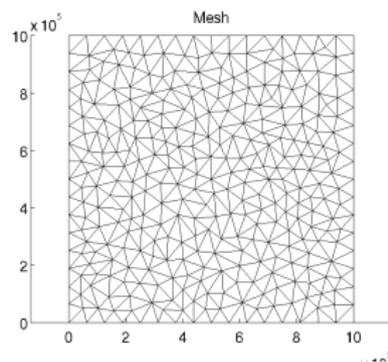
Example: Steady state ice shelf with square domain

- Build mesh over domain with desired resolution (try 100,000 m)

```
1 >> md=triangle(md, 'DomainOutline.exp', 100000);
```

- To plot mesh...

```
1 >> plotmodel(md, 'data', 'mesh');
```



Example: Steady state ice shelf with square domain

- Define glacier type: grounded or ungrounded (default is grounded); typically takes form of...

```
1 >> md=setmask(md, 'floatingicename.exp', 'groundedicename.exp')
```

- For our purposes, set floating ice to 'all'

```
1 >> md=setmask(md, 'all', '');
```

- Call parameterization file (here: use 'Square.par')

```
1 >> md=parameterize(md, 'Square.par');
```

Example: Steady state ice shelf with square domain

- Select ice flow model (here: SSA Shallow-Shelf model; no vertical shear)

```
1 >> md=setflowequation(md, 'SSA', 'all');
```

- Specify platform on which to run the simulation

```
1 >> md.cluster=generic('np', 2);
```

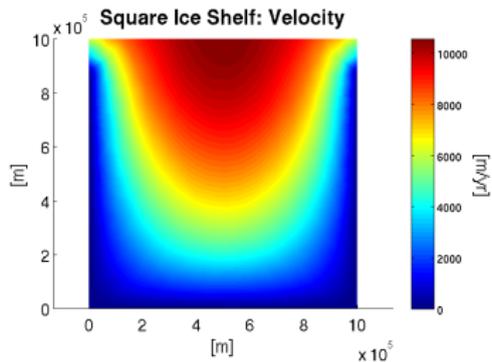
- Compute velocity field for the domain

```
1 >> md=solve(md, StressbalanceSolutionEnum);
```

- Plot the velocity field

```
1 >> plotmodel(md, 'data', md.results.StressbalanceSolution.Vel, ...  
2 'xlabel', '[m]', 'ylabel', '[m]', ...  
3 'title', 'Square Ice Shelf: Velocity');
```

Example: Steady state ice shelf with square domain



- Print the image

```
1 >> print -dpng Velocity.png;
```

Thanks!

